**Vysoká škola báňská – Technická univerzita Ostrava**
**Fakulta elektrotechniky a informatiky**
**Katedra telekomunikační techniky**

# Návrh a realizace síťového analyzátoru na platformě Raspberry Pi

# Design and Implementation of Network Analyzer using Raspberry Pi Platform

**2018** **Bc. Adam Birka**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

# Zadání diplomové práce

Student: **Bc. Adam Birka**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Návrh a realizace síťového analyzátoru na platformě Raspberry Pi
Design and Implementation of Network Analyzer Using Raspberry Pi Platform

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce bude vývoj síťového analyzátoru s displejem na bázi RaspberryPi s podporou funkcí IDS.
1. Proveďte rešerši v oblasti síťových analyzátorů na platformě jednočipových počítačů.
2. Proveďte návrh a realizaci síťového analyzátoru s displejem a akumulátorem na bázi Raspberry Pi.
3. Do návrhu implementujte funkce IDS a dalších síťových funkcí.
4. Navržené řešení otestujte řadou zátěžových testů a výsledky vyhodnoťte.

Seznam doporučené odborné literatury:

UPTON, Eben a Gareth HALFACREE. *Raspberry Pi User Guide*. 4. Wiley, 2016. ISBN 9781119264378.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 30.04.2018

doc. Ing. Miroslav Vozňák, Ph.D.
*vedoucí katedry*

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

prof. Ing. Pavel Brandštetter, CSc.
*děkan fakulty*

# Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *30. dubna 2018*

.....
podpis studenta

# Poděkování

Rád bych poděkoval Ing. Liboru Michalkovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

# Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TUOstrava."

Dne: *30. dubna 2018*

.....
podpis zástupce

# Abstrakt

Cílem této práce je získat co nejvíce poznatků o realizaci handheld zařízení na platformě RaspberryPi. Na základě získaných poznatků takovéto zařízení sestrojit a zprovoznit. Dále do sestrojeného zařízení implementovat síťové funkce, včetně funkce IDS. Navržený model ověřit sadou testů a měřené hodnoty porovnat s teoretickými poznatky. Klíčovým parametrem tohoto měření je IDS, jehož funkce byly ověřovány pomocí testovacích nástrojů.

# Klíčová slova

Raspberry Pi, síťové funkce, systém detekce průniku, handheld

# Abstract

The aim of this work is gaining much knowledge about realization hand-held devices on the platform RaspberryPi. Based on gained knowledge constructthis device and launch. Furthermore implement network functions including IDS function. Designed model verify series of tests and measured merits verify with theoretical knowledge. Crucial parameter of this measure is IDS and its functions were checked via test tools.



# Key words

Raspberry Pi, Network functions, IDS (Intrusion Detection Systems), handheld

# Seznam použitých symbolů

| **Symbol** | Jednotky | Význam symbolu |
|---|---|---|
| **f** | Hz | Frekvence |
| **Vp** | Bit/s | Přenosová rychlost |
| **t** | s | Čas |
| **T** | °C | Teplota |

# Seznam použitých zkratek

| Zkratka | Význam |
|---|---|
| **AP** | Access Point |
| **DoS** | Denial of service |
| **DNS** | Domain Name Systém |
| **FTP** | File Transfer Protocol |
| **HIDS** | Host based IDS |
| **HTTP** | Hypertext Transfer Protocol |
| **HTTPS** | Hypertext Transfer Protocol Secure |
| **HW** | Hardware |
| **ICMP** | Internet Control Message Protocol |
| **IP** | Internet Protocol |
| **IDS** | Intrusion Detection System |
| **LAN** | Local area network |
| **NIDS** | Network based IDS |
| **NOOBS** | New Out Of the Box Software |
| **NSM** | Network security monitor |
| **RAM** | Random access memory |
| **RPi** | Raspberry Pi |
| **SLIP** | Serial Line Internet Protocol |
| **PPP** | Point-to-Point Protocol |
| **SNMP** | Simple Network Management Protocol |
| **SoC** | Systém on chip |
| **SODIMM** | Small outline dual in-line memory module |
| **SSH** | Secure Shell |
| **SW** | Software |
| **TCP** | Transmission Control Protocol |
| **UDP** | User Datagram Protocol |

| | |
|---|---|
| **UTP** | Unshielded Twisted Pair |
| **WAN** | Wide area network |
| **WEP** | Wired equivalent privacy |
| **Wi-Fi** | Wireless fidelity |

# Seznam použitých termínů

| **Termín** | Význam termínu |
|---|---|
| **Bandwidth** | Přenosová rychlost |
| **IDS** | Intrusion Detection Systém - systém detekce průniku. |
| **Subnet** | Podsíť |
| **RPi IDS** | Konkrétní IDS systém implementovaný na platformě Raspberry Pi |

# Obsah

# 1 Úvod

Cílem této práce byl návrh a realizace handheld zařízení a následná implementace síťových funkcí včetně IDS. Největší pozornost byla věnována tvorbě zařízení a následnému testování IDS. Takovéto zařízení dostalo pracovní název RPi IDS. Nejprve se zabývá tato práce teorií IDS a jednočipových počítačů, dále použitím vhodných komponentů pro naše testovací zařízení a volbou vhodného softwaru pro IDS. V další části je zmíněno samotné sestrojení zařízení a následná implementace softwaru a síťových funkcí. Ke konci práce následovalo testování a vyhodnocení výsledků. Jako bonus pro možné budoucí navázání na tuto práci byl přidán i návrh vizuálního prostředí pro IDS systém.

Pro samostatnou realizaci bylo nutné nastudovat teorii. Na základě získaných poznatků byl vybrán vhodný hardware a software. Jako hlavní část zařízení byla vybrána platforma RaspberryPi, na jejímž základě byl sestaven zbytek zařízení. Dalšími klíčovými komponenty byly například 5“ dotykový displej, Rii klávesnice a baterie Adata. Obal pro uložení komponentů byl navržen v programu Autodesk 123D studio a poté vytisknut na školní 3D tiskárně Prusa MK2. Následovala instalace softwaru a implementace síťových funkcí. Za vhodnou distribuci operačního systému byla zvolena linuxová distribuce Raspbian. Na jejím základě byla otestována sada síťových funkcí v příslušných programech. Poslední částí této práce byla implementace IDS a jeho testování. Veškeré výsledy testů byly zapsány do tabulek a následně zpracovány v podobě grafů.

Vše bylo důsledně zdokumentováno včetně vytvořených grafů k jednotlivým testům. V závěru byly porovnány výsledky testů jednotlivých měření a zhodnocena vhodnost vybraných komponentů a softwaru k řešení dané problematiky.

## 1.1 Prezentace problému

Každý rok se objevují nové bezpečnostní hrozby a tradiční nástroje počítačové bezpečnosti přestávají být účinné vůči jejich pokročilejším formám. Za posledních několik let byl zaznamenán nárůst pokročilých bezpečnostních hrozeb, které se už nesoustředí jenom na mezinárodní špionáž a destrukci infrastruktury států, ale jejich obětí se stávají i větší organizace či političtí představitelé. Mezi oběti těchto sofistikovaných útoků se dostávají i menší organizace či jednotlivci. A právě IDS je jedním ze způsobů, jak se takovýmto útokůmbránit pomocí jejich detekce. Na základě této detekce může administrátor najít napadené místo v síti a podniknout kroky k jejímu dalšímu zabezpečení.[4]

## 1.2 Účel a otázky

Hlavním cílem této studie je, kromě realizace IDS, také zkoumání vlivu IDS na výkon za užití jednočipového počítače a porovnání výkonu a schopnosti zpracovávat síťový provoz, konkrétně na RaspberryPi 3 model B. Měření výkonu zařízení bylo provedeno testováním vytížení procesoru a paměti. Na základě naměřených výsledků bylo možné určit nedostatky hardwaru.

**Kladené otázky:**

- Může být RaspberryPi použito jako systém detekce narušení(IDS) v síti?
- Jak obstojí RaspberryPi 3 model B z hlediska propustnosti a výkonu procesoru či paměti?

## 1.3 Limitace

Hlavní předpokládanou limitací realizace systému je hardware, u kterého je možné při realizaci narazit na omezený výkon CPU, pomalé rychlosti MicroSD kartya velikost RAM paměti zařízení RaspberryPi.

## 1.4 Cílová skupina

Navržený produkt je určen pro tři základní cílové skupiny. První cílovou skupinou jsou poskytovatelé internetu, kteří pomocí zařízení mohou analyzovat svou síť a monitorovat aktuální dění v ní. Díky přenositelnosti zařízení tak může technik daného providera analyzovat provoz z jakéhokoli bodu sítě, ke které se připojí. Druhou cílovou skupinou jsou firmy, které chtějí zabezpečit svou firemní síť. Poslední cílovou skupinou jsou školy, které mohou toto zařízení používat pro výuku například počítačových sítí.

# 2 Systém pro detekci síťových útoků

Bezpečnost počítačových sítí nespočívá jen v ochraně proti síťovým útokům a v prevenci, ale je nutná také jejich detekce a následná reakce na tyto útoky. Systém detekce průniku (IntrusionDetectionSystem - IDS) je systém, který odhaluje narušení nebo případné pokusy o narušení bezpečnosti počítačových systémů. Vychází z předpokladu, že úkony narušitele budou na základě přímých i nepřímých indicií odlišitelné od běžné činnosti uživatele. Existuje více metod, kterými lze detekovat narušení, např. analýzou záznamů (logů) systému nebo sledováním provozu v síti.[4][9][17]

## 2.1 Dělení IDS

IDS dělíme podle umístění, principu činnosti a okamžiku zhodnocování.

### 2.1.1 Umístění

Při dělení podle umístění je důležité, odkud IDS získává informace. Z typu informací, které má k dispozici, vyplývá nejen jeho schopnost reagovat na různé druhy útoků, ale také pokrytí hlídané oblasti. Hostitelsky orientované IDS detekují pouze útoky vedené na systém, na kterém jsou provozované.

**HIDS** (Host based IDS)

HIDS jsou IDS orientované na hostitelský systém. Sbírají informace z konkrétního systému. Využívají systémové záznamy generované jádrem a dalšími systémovými zdroji, monitorují probíhající procesy v kontextu s aktivitami uživatelů a změnami v souborovém systému, např. LIDS, Tripwire, Snortinline.

*Výhody*

- Detekce útoků vedených přes šifrovaný kanál (NIDS ze síťové komunikace nic nepozná, HIDS může útok detekovat např. analýzou logů).
- Mohou odhalit útoky, které nelze odhalit síťovými IDS.

*Nevýhody*

- Každý systém musí mít vlastní konfiguraci, s čímž souvisí složitá administrace.
- Může být napaden v rámci útoku na hostitelský systém.
- Může být vyřazen z činnosti útokem DoS.

**NIDS** (Network IDS)

Jedná se o síťový IDS, zpravidla realizovaný na dedikovaných serverech, které zpracovávají informace získané ze síťových rozhraní. Důležité je jejich umístění, aby zachycovaly co největší (ideálně všechen) síťový provoz, např. Snort, Bro, Tamandua.

*Výhody*

- Dobře rozmístěné senzory mohou monitorovat velkou síť.
- Systém neovlivňuje provoz sítě.

*Nevýhody*

- Obtížné zpracování všech paketů v sítích s velkým provozem.
- Omezení použití v přepínaných sítích (nutno použít SPAN porty přepínačů).
- Nemůže analyzovat šifrovaný provoz.
- Nelze jednoznačně určit, zda byl započatý útok úspěšně dokončen.

**Hybrid IDS**

Jedná se o komplexní systém kombinující oba předchozí typy, např. Prelude, OSSIM.

*Výhody*

- Výhody NIDS a HIDS dohromady.

*Nevýhody*

- Obtížnější nasazení a administrace systému.

### 2.1.2 Princip činnosti

Důležitou vlastností IDS je princip činnosti při detekování útoků. Určuje např. schopnost rozpoznávat dosud neznámé typy útoků nebo míru generovaných falešných varování.

**Porovnávání vzorů**

Jedná se o detekci podle srovnávání se vzory (pravidly, signaturami). Ty odpovídají událostem nebo posloupnosti událostí typických pro známé útoky nebo narušení systémů. Vzory jsou vytvářeny na principech konečných automatů, rozhodovacích stromů, expertních systémů, neuronových sítí, pravděpodobnostních modelů atd.

*Výhody*

- Snadné psaní vzorů (např. konkrétně pro Snort).
- Existence vzorů pro většinu známých útoků.

*Nevýhody*

- Možnost maskování škodlivé aktivity a oklamání senzoru.

**Detekce statistické anomálie**

Upozorní na podezřelé odchylky od dlouhodobým sledováním stanoveného normálního chování sítě. Např. sleduje se delší dobu (může být i v závislosti na denní době) počet dotazů na DNS server. Detekce zvýšeného počtu dotazů pak může znamenat útok.

*Výhody*

- Detekce neznámých útoků.

*Nevýhody*

- Možnost vzniku mnoha falešných poplachů z důvodu nepředvídatelného chování uživatelů.
- Potřeba dlouhého zkušebního provozu, než systém nasbírá dostatek informací pro stanovení normálu.

**Korelační**

Vyhledávají souvislostí mezi jevy probíhajícími na několika místech. Příkladem může být korelace mezi SW a operačním systémem (pokud je detekován útok na IIS web server a na serveru běží Apache s Linuxem, varování může být zahozeno) nebo korelace mezi Nessusem (bezpečnostní skener) a Snortem (Snort detekuje útok na přetečení zásobníku a Nessus zjistil aplikaci zranitelnou tímto druhem útoku, varování je pak generováno s vysokou prioritou).

*Výhody*

- Menší pravděpodobnost falešného poplachu.

*Nevýhody*

- Složitější nasazení systému.

### 2.1.3 Okamžik vyhodnocování

Okamžik vyhodnocování určuje rychlost detekování útoku. Vyhodnocování v reálném čase je sice výkonnostně náročnější, ale umožňuje okamžitou reakci na útok.

**V reálném čase**

Průběžné sbírání a vyhodnocování informací. Systémy umožňují generovat real-time poplachy a automatické odezvy na útok.

*Výhody*

- Bezprostřední reakce.
- Rychlejší zotavení po útoku.
- Efektivnější způsob identifikace a usvědčení útočníka.

*Nevýhody*

- Výkonnostně náročné při větším provozu.

**Dávkově orientovaný přístup**

Periodický sběr informací s následným vyhodnocováním. Vhodné pro prostředí s poměrně nízkými riziky a potenciálními ztrátami.

*Výhody*

- Malé zatížení systému.
- Snadno odhalitelný útok na stejný cíl (lze vyhodnocovat informace nasbírané za delší časové období).
- Možná manuální analýza.

*Nevýhody*

- Není možná bezprostřední reakce.
- Velký objem ukládaných dat.

# 3 Možná realizace IDS

## 3.1 RaspberryPi

RaspberryPi je britský miniaturní počítač, který je založen na SoC společnosti Broadcom. Disponuje HDMI výstupem, síťovým rozhraním, USB porty a především řadou GPIO rozhraní. Je určen pro domácí automatizaci, experimentální elektroniku a výuku na školách. Uživatelé pro něj našli ale i další uplatnění a používají jej jako multimediální centrum, domácí server či jako desktopový počítač.[3]

Celý návrh desky je uvolněn pod svobodnou licencí, stejně jako software pro počítač určený. Primárním operačním systémem je Raspbian, klon linuxové distribuce Debian, který je přeložen pro procesory ARM. Vznikla však řada dalších systémů včetně OpenWRT, NetBSD, FreeBSD, Gentoo a dalších.[8]

### *RaspberryPi 3 model B*

RaspberryPi 3 model B je třetí generací RaspberryPi se zcela novým 1,2GHz 64-bitovým 4-jádrovým procesorem ARM, 1GB RAM a integrovanou WiFi (b/g/n) a Bluetooth4.1. RaspberryPi 3 si zachovává stejné rozměry a rozmístění konektorů jako jeho předchůdci, takže je plně kompatibilní se všemi stávajícími moduly. Na desce najdeme jen několik drobných rozdílů.[10]

*Hlavní změny oproti předchozím generacím*

- Nový 1,2GHz 64-bitový 4-jádrový procesor ARM Cortex-A53
- Rychlejší VideoCore @ 400 MHz pro video / 300 MHz pro 3D grafiku
- Integrovaná WiFi 802.11 b/g/n
- Integrovaný Bluetooth 4.1 LE
- Kontrolní diody a RUN kontakty musely uvolnit místo Wifi/Bluetooth anténě. Kontrolky se proto přesunuly na druhou stranu od DSI portu a RUN kontakty k opačnému konci GPIO.
- Zacvakávací microSD slot byl nahrazen jednodušším zasouvacím microSD slotem.
- Přesun zaznamenaly i některé drobné komponenty na desce.
- Nově se jako minimum pro zdroj určuje ~1,3A zdroj.

**Výkon**

Ve srovnání s RaspberryPi 2 má procesor o 33% vyšší frekvenci (1,2GHz vs 0,9GHz) a modernější jádro s efektivnější instrukční sadou, speciálně při provádění operací s 64-bitovými hodnotami. Video a 3D výkon také narostl, jak se zvýšila frekvence VideoCore na 400 MHz (z 250 MHz) pro zpracování videa, respektive na 300 MHz (z 250 MHz) pro zpracování 3D grafiky.Díky procesoru ARMv8 je možné na RaspberryPi 3 spustit kompletní škálu ARM GNU/Linux distribucí.

*WiFi a Bluetooth*

Největší změnou je integrace WiFi 802.11 b/g/n a Bluetooth 4.1 LE, dalších dvou možností, jak připojit Raspberry k LAN, přímo na desku. Ovladače pro Bluetooth a WiFi jsou součástí posledního NOOBS 1.8.0.

*GPIO*

Rozložení GPIO headeru zůstalo beze změn. Pi 3 má stejných 40 pinů jako Pi 2B, 1B+ a 1A+. Je tak zajištěna 100% kompatibilita se stávajícími rozšiřujícími moduly a HAT deskami.

*Obrázek 3.1: Raspberry Pi 3 Model B*

## 3.2 Další alternativy k RPi

Mezi podobná, či konkurenční zařízení pro RaspberryPi lze zařadit například Arduino, Banana Pi, OrangePi či Pine64.

### 3.2.1 Arduino

Jedná se o fyzickou open-source počítačovou platformu, jež je založena na jednoduché mikrokontrolní desce.Vývojářské prostředí slouží k zápisu softwaru. Pomocí Arduina lze vyvíjet interaktivní předměty, získávat vstupy od různých spínačů a senzorů a ovládat například světla, motory či jiné fyzické výstupy. [13]

Nevýhody: Slabší výkon, složitější konfigurace

Výhody: Vhodné pro senzorová využití, dostupná cena

### 3.2.2 Banana Pi M3

BananaPi R1 je malý jednodeskový počítač, který je schopen zvládnout funkci NASu, routeru, Wi-Fi AP i domácího serveru. Běží na něm běžná linuxová distribuce, do které je možno doinstalovat libovolný software. Díky výkonnému dvoujádrovému procesoru ARM Cortex-A7 s integrovaným GPU jádrem Mali400MP2 není nutné se omezovat na úsporné serverové aplikace.[14]

Nevýhody: Malá podpora, slabá komunita vývojářů

Výhody: Vyšší výkon, Gigabitový Ethernet, dostupná cena

### 3.2.3 Orange Pi

Orange Pi je další z řady jednodeskových počítačů. Tvůrci Orange Pi připravili sérii minipočítačů vycházejících z návrhu RaspberryPi, postavených na platformě vícejádrovýchSoC Allwinner ARMv7. Základní modely jsou tři, přičemž dva používají dvoujádrovýAllwinner A20 s grafickým jádrem Mali-400 a třetí čtyřjádrovýAllwinner A31 s grafikou PowerVR SGX544MP2. Vedle gigabitového Ethernetu je také možnost připojení k sítí pomocí Wi-Fi 802.11b/g/n, která původním RaspberryPi 1 a Raspberry Pi 2 chybí. Dále je Orange Pi vybaven dvojicí USB 2.0 portů u nejlevnější verze, a čtveřicí u dražších verzí. Samozřejmostí je HDMI výstup či GPIO piny pro vývojáře. Výbavu doplňuje 1 GB RAM a připraven je i slot pro paměťové karty typu micro SD.

Nevýhody: Malá podpora, slabá komunita vývojářů

Výhody: Vyšší výkon, Gigabitový Ethernet, dostupná cena

### 3.2.4 Pine A64+

Pine A64+ se v mnoha směrech podobá oblíbenému RaspberryPi 3. Má 64bitový procesor Cortex A53 CPU vycházející z architektury ARM, slot pro MicroSD karty či porty pro Ethernet a HDMI. Pine 64+ má však lepší grafiku díky podpoře procesoru ARM Mali 400 MP2 – ten dovoluje zpracovávat i 4K video.

Nevýhody: Malá podpora, slabá komunita vývojářů, velké rozměry

Výhody: Vyšší výkon, Gigabitový Ethernet, dostupná cena

### 3.2.5 Raspberry Pi Compute Module 3

Compute Module 3 zahrnuje čtyřjádrový 64bitový procesor Broadcom BCM2837 a 1 GB paměti RAM, což je shodné jako u dosavadního nejvyššího modelu Raspberry Pi 3, oproti němu však má poloviční velikost a navíc u něho chybí klasické komunikační rozhraní jako

Ethernet, Wi-Fi modul, USB, SD Card a port pro připojení displeje.Signály pro chybějící porty se společně objevují v novém tzv. edge konektoru, který pasuje do socketu SODIMM, jenž se obvykle využívá v noteboocích kvůli možnostem upgradu paměťových modulů.V důsledku to znamená, že se nový Raspberry Pi může vestavět do různých robotů, průmyslových strojů a podobně.[8]

Nevýhody: Vyšší cena, nutnost zapojení modulu do základní desky.

Výhody: Vyšší výkon, možnosti v zapojení do příslušné základní desky

## 3.3 Výběr vhodného operačního systému

### 3.3.1 Raspbian

Je operační systém odvozenýz Debianu pro RaspberryPi i osobní počítače.Je oficiálně poskytován nadací RaspberryPiFoundation jako primární operační systém pro single-board počítače z rodiny RaspberryPi.Raspbian byl vytvořen Mikem Thompsonem a Peterem Greenem jako nezávislý projekt.První sestavení tohoto systému bylo dokončeno v červnu 2012. Systém je i nadále v aktivním vývoji, Raspbian je vysoce optimalizovaný pro ARM procesory užité v RaspberryPi.Raspbianpoužívá jako hlavní desktopové prostředí PIXEL. Obsahuje přes 35 000 balíčků předkompilovaného software k okamžité instalaci. Raspbianje nejčastěji doporučovaný operační systém pro RaspberryPi.

### 3.3.2 Kali Linux

Kali je linuxová distribuce zkompletovaná na míru penetračním testerům. Jedná se o operační systém se všemi základními nástroji, potřebnými k síťovým útokům a bezpečnostním testům. Další výhodou připravené distribuce je, že díky komunitě a značnému počtu uživatelů se snižuje riziko zavlečení nechtěného kódu při bezpečnostních testech.

Kalilinux vychází z linuxové distribuce. Předchůdcem Kali Linuxu byl BackTrack. První BackTrack byl dostupný v květnu 2006. Postupně vycházel až do verze 5, to bylo někdy na konci roku 2011. Kalilinux a i jeho předchůdce jsou dílem stejné komunity.

### 3.3.3 Ubuntu

Ubuntu je linuxová distribuce postavená na distribuci Debian. Zaměřuje se na desktopové uživatele, kterým se snaží nabídnout co nejpohodlnější instalaci, moderní prostředí a jednoduché ovládání. Nová verze vychází každých šest měsíců a má podporu devět měsíců. Jednou za dva roky pak vyjde takzvaná LTS verze s pětiletou podporou.

### 3.3.4 Security Onion

Jedná se linuxovou distribuci pro IDS systémy a monitorovací systémy síťové bezpečnosti (NSM - network security monitor). Je založen na základě linuxové distribuce Ubuntu a obsahuje celou řadu bezpečnostních nástrojů. Díky jednoduchosti jednotlivého

nastavení monitorovacích sensorů a bezpečnostních nástrojů se stal tento systém oblíbeným právě pro systémy IDS.

## 3.4 Možné SW nástroje pro realizaci a testování IDS

### 3.4.1 VMware Workstation

Program VMware Workstation je emulátorem, pod kterým lze vytvořit virtuální hardware, a na něm nainstalovat a provozovat další operační systém, a to za běhu Windows. Emulace pomocí VMware Workstation je tak dokonalá, že má simulovaný operační přístup i k lokální síti a připojení k internetu, zvukové kartě a optickým mechanikám počítače, na kterém je simulován. Simulace operačního systému se hodí např. pro sledování následků různých zásahů do systému, testování virů bez ohrožení běžně používaného systému, nebo současný běh Windows a Linuxu.

### 3.4.2 Snort

Snort je program, patřící do kategorie síťových IDS, založených na pravidlech. Více o něm v kapitole 3.5 Snort.

### 3.4.3 Iperf

Tento program je vhodný pro testování možností sítě. Program zahltí síť TCP (Transmission Control Protocol) nebo UDP (User Datagram Protocol) pakety a tím reprezentuje přenos dat po síti. Program funguje ve dvou režimech - v serverovém nebo klientském. Po navázání spojení začne program měřit a vypisovat propustnost mezi serverem a klientem. V programu lze nastavit čas měření, interval sdělování výsledku měření a mnoho dalších parametrů, jako jsou velikost TCP segmentů, simulace více klientů, velikost TCP okna či délka vyrovnávací paměti.

### 3.4.4 Wireshark

Wireshark (dříveEthereal) je protokolový analyzátor a paketový sniffer. Wiresharknabízívelice podobné funkcejakotcpdump, navíc však obsahuje grafické uživatelské rozhraní a mnoho možností uspořádání a filtrování zobrazených informací. Aplikaceumípřepnoutsíťovou kartu do promiskuitního režimu a díky tomu dokáže zachytávatveškeroukomunikaci na připojeném médiu.

### 3.4.5 Suricata

Je analyzátor síťového provozu.Suricata je zcela zdarma (pod open-source licencí GNU GPLv2) a představuje alternativu k řešení Snort (také open-source), jež je mezi uživateli IDS známější. Oba mají společné to, že zpracovávají jim dostupné pakety, srovnávají je s předem nastavenými pravidly a na základě shody s některým z nich zjišťují nestandardní komunikaci, podezřelou aktivitu či dokonce pokusy o útok. Tyto události jsou zaznamenávány a je možné si

nechat zaslat upozornění, nebo dokonce analyzátoru umožnit změnit pravidla firewallu a připojení zablokovat.

### 3.4.6 Sar

SAR je nástroj na sledování výkonu, který dokáže zobrazit nejen současná, ale i starší měřená data. Prostřednictvím SAR lze sbírat data a připravovat statistiky za dlouhé časové období.

### 3.4.7 Htop

Htop je interaktivní nástroj zobrazujícíběžící procesy a vytíženost procesoru a paměti v reálném čase. Disponuje přehlednounabídkou, a umínapříkladvyhledatkonkrétní proces stisknutím klávesy. Ovládání programu je možné i pomocí kurzoru myši.

### 3.4.8 Hping

Hping3 je síťový nástroj schopný odesílat vlastní pakety TCP/IP a zobrazovatcílové odpovědipodobnějako program ping s odpověďmi ICMP. Díky němu lze provádět penetrační testy. Má mnoho nastavení, která se uvádějí za příkaz hping3. Jejich výpis je níže.

Příklad:

hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --flood--rand-source (ip adresa oběti)

- flood - posílá pakety co nejrychleji a nezobrazuje odpovědi

- rand-dest - náhodný adresový režim

-V - Verbose mód

-c - počet paketů

-d - velikostdat

-S - nastavení příznaku SYN

-w -velikost okna

-p - cílový port

-s - základní zdrojový port

### 3.4.9 Nmap

Nmap je jeden z nejpopulárnějšíchnástrojů pro skenováníportů. Má mnoho nastavení a hodí se k základnímprověrkám bezpečnosti, zejménapak po různých úpravách firewallu nebo konfiguracesítě.

### 3.4.10 host

Host je jednoduchý nástroj pro vyhledávání DNS. Obvykle se používá k převodu jmen na IP adresy a naopak.

### 3.4.11 Nessus

Nessus Professional je nejrozšířenější odhadové rosení identifikace slabých míst, která útočnici používají k proniknuti do počítačové sítě. Nessus nabízí efektivní a komplexní balíček testovacích skenů.

## 3.5 Snort

Snort je open source NIDS (network intrusiondetection system, systém pro síťovou detekci průniků) a NIPS (network intrusionpreventionsystem, systém pro síťovou blokaci průniků), který vytvořil Martin Roesch v roce 1998. Snort umožňuje zachytávání paketů pomocí knihovny libcap a jejich následnou analýzu proti množině pravidel. Jedná se tedy o signature-based bezpečnostní řešení. V současné době se jedná o nejčastěji instalované řešení IDS na světě. Snort může běžet ve třech různých módech: sniffer mode (režim slídiče), packetlogger mode (režim záznamníku) a network intrustiondetectionsystem mode (režim detekce narušení).

Architekturu Snort lze rozdělit na čtyři části: sniffer (zachytávání paketů), preprocessor, detectionengine (detekci) a output (výstup). Tyto části na sebe postupně navazují.

Cílem snifferu (korektněji packetsnifferu) je zachytávat a ukládat datové pakety putující v síti, ať již dočasně nebo trvale. Sniffery existují často jako samostatný software obsahující i komplexní nástroje pro grafické zobrazování obsahu a analýzu použitých aplikací a protokolů. Sniffer může být umístěn na cílovém zařízení a zachytávat data pro něj určená, nebo získávat data z jiné části sítě. Ve druhém případě je zdrojem dat aktivní (port mirroring) nebo pasivní síťový prvek. Sniffer kromě již zmíněných bezpečnostních aplikací slouží také pro detekci technických problémů a jejich odstraňování. Na druhé straně se jedná o jeden ze základních nástrojů pro tvorbu a provádění síťových útoků. Zachytávání dat uprostřed sítě se mimo jiné z toho důvodu stává problematické kvůli stále častějšímu šifrování dat na síťové vrstvě.

Úkolem preprocessoru je normalizace a dekódování protokolu na vyšší vrstvě ISO/OSI modelu. To umožňuje přesnější a výkonnější porovnávání průchozích paketů s použitými pravidly. Preprocessor na jednotlivých vrstvách skládá jednotlivé příchozí pakety do proudu dat.

Samotné srdce IDS/IPS tvoří detectionengine. Detection engine přijímá data od preprocessorů a porovnává je postupně s databází pravidel. Na základě shody je poté pro každý paket nebo proud dat stanoven předem daný scénář. Ten zahrnuje blokaci paketu, zaznamenání a informování další vrstvy programu, uložení pro další analýzu. Aktualizovaná databáze bezpečnostních hrozeb je alfou a omegou každého signature-based IDS/IPS a spolu s výkonem zpracovávání patří stejně jako například u antivirových řešení k rozhodujícím faktorům

vypovídajícím o kvalitě daného řešení. Funkci programu a jeho chování je nutné kontrolovat a vyhodnocovat. K tomu slouží poslední funkční vrstva – výstup (output). Kromě standartního logování událostí do textového souboru umožňuje Snort generovat SNMP (simple network management protocol) zprávy nebo zapisovat do databáze.[1][9]

*Stažení a instalace*

Software Snort je možné získat například z oficiálních webových stránek projektu. K dispozici jsou verze pro operační systém Microsoft Windows, Linux a zdrojové kódy. Nainstalovat jej lze na většině linuxových distribucí také pomocí stažení z repositáře.

### 3.5.1 Aktualizace pravidel

Nejdůležitější částí korektní funkce IPS/IDS je sada pravidel pro detekci útoků. Tvůrce software Snort, společnost Sourcefire, nabízí placenou verzi aktuálních pravidel ve formě ročního předplatného. Tato pravidla jsou vytvářena odborníky v reakci na vznik nových hrozeb. Volně dostupná výše uvedená pravidla jsouk dispozici taktéž zdarma ke stažení stříceticidenním zpožděním. Samostatně lze získat i aktuální pravidla vytvářená komunitou uživatelů.[18]

### 3.5.2 Režimy Snortu

*Sniffer mode*

- Tento mód zachytává pakety (sniffuje) procházející sítí a zobrazuje je uživateli na obrazovce.

*Packetlogger mode*

- Logger je v podstatě rozšířený sniffer mode, rozdíl je v tom, že se paketová data nebo hlavičky zaznamenávají do log souborů na pevný disk.

*Network intrustiondetectionsystem mode*

- Nejvýznamnější režim programu, Snortzde odchytává síťová data a analyzuje je v kontextu s uživatelem definovanými pravidly a provádí akce podle nálezu.

*Inline*

- Získává pakety z iptables. Na základě pravidel rozhoduje o zahození nebo povolení paketů. V tomto módě pracuje jako HIDS.

### 3.5.3 Komponenty Snortu

Snort je logicky rozdělený do několika komponent. Tyto komponenty pracují společně, detekují tak jednotlivé útoky a generují výstup v požadovaném formátu. IDS Snort se skládá z následujících hlavních komponent:

- Jednotka paketového záchytu
- Zásuvné moduly preprocesoru
- Detekční jednotka

- Systém logování a výstrah
- Výstupní zásuvné moduly

### *Jednotka paketového záchytu*

Komponenta bere pakety z různých síťových rozhraní počítače a předzpracovává pakety pro zaslání detekční jednotce. Rozhraním může být Ethernet, SLIP, PPP atd.

### *Zásuvné moduly preprocesoru*

Preprocesory jsou komponenty, které lze použít ve Snortu k uspořádání nebo úpravě datových paketů předtím, než detekční jednotka provede operace, aby zjistila, jestli je paket generován vetřelcem. Některé preprocesory také hledají neobvyklosti v hlavičce paketu. Preprocesory jsou velmi důležité pro NIDS k přípravě datových paketů pro analýzu v detekční jednotce. Hackeři užívají různou techniku a různé způsoby k oklamání IDS.Preprocesory se také používají pro paketovou defragmentaci.

### *Detekční jednotka*

Detekční jednotka je nejdůležitější část Snortu. Jejím úkolem je systematicky porovnávat data uvnitř každého paketu, zda obsahuje zvláštní řetězec nebo hodnotu sdruženou s nějakým pravidlem. Detekční jednotka pro tento účel využívá pravidla Snortu. Pravidla jsou načítána do vnitřních datových struktur nebo řetězců a jsou porovnávána proti všem paketům. Jestliže paket odpovídá některému pravidlu, vyvolá se příslušná akce. Příslušnou akcí zde může být zápis do logu nebo vytvoření výstrahy. Detekční jednotka je časově náročný modul Snortu. Hodně proto záleží na tom, jak je počítač výkonný a kolik pravidel má definovaných. Jestliže je zatížení sítě příliš vysoké a Snortpracuje v NIDS režimu, může docházet k zahazování některých paketů, což vede k nesprávné odezvě v reálném čase. Zatížení detekční jednotky je ovlivněno následujícími faktory:

- počet pravidel
- výkon počítače, na kterém Snort právě běží
- rychlost vnitřní sběrnice, používané v počítači, kde běží Snort
- zatížení sítě.

### *Systém logování a výstrah*

Tato část Snortu závisí na tom, co detekční jednotka najde podezřelého uvnitř paketu, což je následně použito k záznamu do logu nebo k vytvoření výstrahy. Logy mají strukturu jednoduchého textového souboru v tcpdump tvaru nebo v jiných formách. Všechny logovací soubory se implicitně ukládají do adresáře /var/log/snort.

### *Výstupní zásuvné moduly*

Výstupní jednotka nebo zásuvný modul provádí operace podle toho, jak chceme mít uloženy výstupy vytvořené systémem logování a výstrah. S ohledem na nastavení, mohou výstupní moduly dělat následující věci:

- Zaznamenávat do /var/log/snort/alert souboru (nebo nějakého jiného)

- zasílání SNMP trapů
- zasílání zprávy do syslogu
- zapisovat do databáze jako MySQL nebo Oracle
- generovat XML výstup
- modifikovat konfigurace routerů a firewallů.

### 3.5.4 Pravidla Snortu

Jednou z nejlepších vlastností Snortu je možnost snadného vytváření a přidávání pravidel (obrázek 3.1). Jednotka pravidel programu Snort poskytuje rozšířený jazyk, který umožňuje napsat vlastní pravidla, která je možné přizpůsobit potřebám sítě. Každé pravidlo se skládá ze dvou částí: hlavičky a volby (options). Hlavička pravidla obsahuje akci pro vykonání, protokol, ke kterému se pravidlo vztahuje, zdrojové a cílové adresy s porty. Options pravidla nám dovolují vytvořit popisnou zprávu spojenou s pravidlem a kontrolovat různé druhy dalších atributů paketu, které Snort používá v rozsáhlé knihovně zásuvných modulů.[18]

*Snort poskytuje několik vestavěných akcí, která lze použít pro vytvoření pravidel*

- *Pass* (předání) – ignoruje paket.
- *log* (zaznamenání) – zaznamená paket.
- *alert* (výstraha) – vygeneruje výstrahu a paket zaznamená.
- *activate* (aktivace) – vygeneruje výstrahu a vyvolá k otestování další pravidlo.
- *dynamic* (dynamika) – akce „dynamic“ jsou vyvolávány pouze dalšími pravidly užitím akce „activate“. Za normálních okolností nejsou používány na paket.
- *drop* (zahození) – přidá do iptables pravidlo pro zahození paketu a paket zaznamená.
- *reject* (odmítnutí)– přidá do iptables pravidlo pro zahození paketu, paket zaznamená a pošle TCP reset, jestliže je protokolem TCP nebo ICMP zprávu o nedostupnosti portu, jestliže je protokolem UDP.
- *sdrop* - přidá do iptables pravidlo pro zahození paketu, ale paket nezaznamená.

Implicitně se jako první testují pravidla Aktivace, poté pravidla Dynamiky, následují pravidla Výstrahy, poté přijdou pravidla Předání a končí pravidly Zaznamenání. Nicméně se dá pořadí měnit.


Obrázek 3.2: *Tvorba pravidel snortu*

## 3.6 Bro IDS

BRO je open-source NIDS založený na Unixu, který pasivně monitorujeprovoz a hledá podezřelé aktivity. BRO pracuje tím způsobem, že nejdříverozebere komunikační provoz, aby extrahoval sémantiku aplikační vrstvy a v nídále hledá objevující se náznaky podezřelých aktivit. Tyto analýzy jsouschopny zachytit jak již známé hrozby na základě signatur, známých okolností nebo událostí, tak i očekávané hrozby na základě odepřeného přístupu čispolehlivosti spojení určité služby. Jeho bezpečnostní skripty jsou psányvlastním Bro jazykem a jsou schopny spouštět i některé akce. V případě, že siuživatel osvojí jazyk Bro, má možnost psát skripty vlastní, popř. upravovatskripty stávající. Bro obsahuje velké množství již hotových skriptů, které jsoupřipraveny k použití a k nimž potřeba znalost tohoto jazyka. Tyto skriptyjsou schopny zachytit téměř všechny známé hrozby, a to s nízkým FPR.

# 4 Realizace IDS

## 4.1 Sestrojení handheld zařízení

Prvním krokem, který bylo potřeba realizovat, bylo samotné zařízení. Název handheld zařízení byl pro jednodušší popis dále v práci RPi IDS. Požadavky pro handheld zařízení byly následující:

- obal vytvořený na 3D tiskárně
- vhodný hardware
- funkční akumulátor
- vstupní ovládací zařízení, respektive klávesnice, touchpad
- výstup obrazu na vlastním LCD displeji

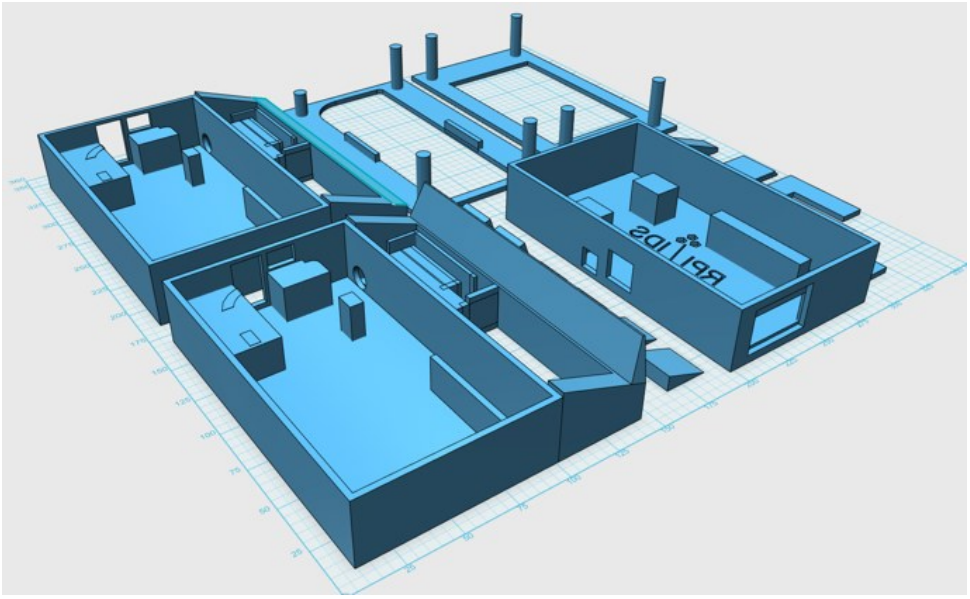
Tabulka 4.1: Zvolené komponenty RPi IDS

| | Hlavní komponenty RPi IDS |
|---|---|
| Základní deska | Raspberry Pi 3 model B |
| Displej | 5" displej s rozlišením 800x600 |
| Baterie | Baterie Adata PV120 |
| Klávesnice | Bezdrátová klávesnice Rii K3 |
| Uložiště | Samsung SDXC 32GB |
| Obal | Vytištěný obal na míru |

Po výběru vhodných komponentů, popsaných v teoretické části, bylo potřeba provést samotný tisk obalu. Následně byly komponenty zakomponovány do vytisknutého obalu.

### 4.1.1 Realizace obalu

Pro návrh obalu byl využit program od firmy Autodesk, a to 123D Design studia. Nejprve bylo potřeba naměřit velikosti a vytvořit si náčrt pro přenesení daných mír do programu. Dále v programu pomocí 3D objektování byl realizován samotný návrh pro 3D tisk. Návrhy byly celkem dva. První testovací, na jehož základě bylo potřeba zjistit možnosti a přesnost 3D tiskárny. A druhý finální, který byl sestaven již na základě poznatků z prvního testovacího tisku.



Obrázek 4.1: *Finální návrh 3D obalu*

### 4.1.2 Tisk/výroba

Samotný tisk probíhal na 3D tiskárně Prusa3D MK2. Návrh vytvořený v programu 123D Design byl exportován do formátu .stl, aby s ním mohl dále program Prusa3D Slic3r MK2 pracovat. V programu Prusa3D Slic3r MK2 byl převeden načtený soubor do formátu .gcode, který se dále nahrává do samotné tiskárny a ta na jeho základě tiskne požadovaný objekt. Při tvorbě .gcode souboru je nutno správně zadat nastavení tisku. Konkrétně je zde potřeba nastavit kvalitu neboli přesnost tisku na milimetry, která se odráží i na délce tisku, dále tiskový materiál a typ tiskárny.

Obrázek 4.2: *Finální obal vytištěný na 3D tiskárně*

U finálního obalu došlo k realizaci širších stěn, dopočtu některých měr a vyřešení napájení na základě magnetického konektoru.

Jak lze z obrázků vidět, zařízení je rozděleno na dvě části, spodní a vrchní. Ve spodní části se nachází baterie, na níž je usazena klávesnice. Napájecí kabel je dále veden v zadní části s magnetickým výstupem konektoru pro připojení vrchní části. Vrchní část obsahuje samotné Raspberry Pi s displejem. Po usazení vrchní části do určeného místa dojde k propojení magnetického konektoru, umístěného ve spodní části, s konektorem napájení základní desky umístěné ve vrchní části. Dojde tak k okamžitému spuštění zařízení.

## Operační systém

Po realizaci hardwarové části bylo potřeba pro rozchození IDS systému nainstalovat na zařízení operační systém Raspbian. K tomu byl využit nástroj noobs od tvůrců Raspberry, vytvořený pro zjednodušení instalace OS na paměťovou kartu. Po úspěšné instalaci operačního systému byl proveden upgrade na nejnovější verzi systému.

## 4.2 Instalace potřebných nástrojů

Pro realizaci IDS bylo nutno nainstalovat následující nástroje:

- snort
- wireshark,
- python-tk
- iperf
- sar
- htop
- hping či
- glade

Jelikož zařízení pracuje s operačním systémem Raspbian založeném na linuxové distribuci Debian, bylo možné doinstalovat některé balíčky z repositáře. Takovouto instalaci lze provést v případě snortu pomocí příkazů:

```
apt-get update
apt-get install snort
```

Obdobný zjednodušený postup instalace byl realizován i v případě dalších programů jako. V tomto případě se jednalo o příkaz:

apt-get install (název programu, či zkratky)

Místo textu v závorce lze uvést název programu, či zkratku názvu, pod kterou se v repositáři nachází.

## 4.3 Konfigurace IDS

Pro otestování funkcí IDS byly zvoleny nejprve virtualizace. Spuštěny byly v programu VMwarecelkem 4 virtuální stroje s linuxovýmidistribucemiKali, Ubuntu, Raspbian a SecurityOnion.

### 4.3.1 Realizace IDS pomocí Snort

Po zmíněné instalaci programu Snort, bylo potřeba pomocí editace souboru snort.conf specifikovat naši (domácí) a externí síť. Domácí síť v souboru je pojmenovaná jako HOME_NET a externí síť jako EXTERNAL_NET. Domácí síť byla specifikována subnetem 192.168.0.0/24. Její rozsah činí tedy 256 IP adres, z čehož první adresa je adresa sítě a poslední je broadcastová adresa. Možných hostů v síti je tedy 254. Pro externí síť byla použita negace domácí sítě, tím pádem jakákoli adresa mimo domácí síť se zaznamená jako externí síť. Dále bylo nutno přidat či odebrat implementaci souborů s pravidly. Pro testovací účely byla

vytvořenakromě již definovaných pravidel vlastní pravidla v souboru myrules.rules, která byla definována v souboru snort.conf.

### 4.3.2 Tvorba pravidel pro IDS

Ačkoli pro většinu testů byly použity pravidla stážená v balíčku snort3-communit-.rules.tar.gz z webových stránek snort.org, nabízí se možnost tvorby vlastních pravidel. Pro vlastní tvořená pravidla byl v této práci vytvořen soubor myrules.rules, ve kterém byla dále pravidla specifikována. Tvorba pravidel je popsána na příkladu:

Pravidlo č.1:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Prichozi FTP
spojeni"; flags:S; sid:10000;)
```

První pravidlo začíná slovem alert, které určuje, že se při zachycení packetu splňujícího dané pravidlo, vypíše zpráva do konzole a také uloží do souboru, tzv. logu, nacházejícího se v adresáři /var/log/snort.Pravidlo pohlíží dále, zda se jedná o tcp paket pocházející z externí sítě jakéhokoli portu směrujícího do domácí sítě s portem 21, respektive portem určeným pro FTP spojení. Určená zpráva pro výpis je zde "Prichozi FTP spojeni " a identifikační číslo sid: 10000. Pokouší-li se někdo navázat FTP spojení do domácí sítě HOME_NET, IDS na to upozorní.

Pravidlo č.2:

```
alert tcp any any -> any any (msg:"Nalezen obsah slova:
terorismus"; content:"terorismus"; nocase; sid:10003;)
```

Druhé pravidlo vypíše do konzole a zaznamená upozornění výskytu packetu splňující pravidlodo logu na základě tcp paketu pocházejícího z jakékoli sítě jakéhokoli portu směrujícího do jakékoli sítě jakéhokoli portu. Nutno podotknout, že záleží na zapojení IDS, zda takovýto packet zachytí. Nelze tedy zachytávat packety v libovolné síti, kde například nemá IDS spojení. V případě, že packet obsahuje slovo "terorismus" vypíše se zpráva: "Nalezen obsah slova: terorismus". Nezáleží na velikosti písmen a identifikační číslo pravidla bylo stanoveno sid: 100003. Výsledkem je tedy to, že když někdo vyhledává v sítí slovo terorismus, IDS o tom informuje.

Pravidlo č.3:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"V siti
probiha ping test"; icode:0; itype:8; sid:10002;)
```

Třetí pravidlo začíná opět slovem alert, jehož funkci je zmíněna v předchozích pravidlech. Dále však sleduje icmppackety pocházející z externí sítě směrující do domácí sítě. ICMP pakety využívá například program Ping. Program Ping využívá ICMP paketů k prověřování funkčnosti spojení mezi dvěma síťovými zařízeními. V případě, že takovýto paket

poputuje z externí sítě do domácí, bude detekován a IDS informuje o této skutečnosti zprávou "V siti probiha ping test".

Pravidlo č.4:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"Nalezen obsah
slova ufo"; content:"ufo";nocase; sid:10004;)
```

Čtvrté pravidlo je obdobné druhému, ovšem detekuje packety směrující z domácí do externí sítě a sleduje obsah slova "ufo". Při detekci vypíše větu "Nalezen obsah slova ufo". Stejně jako u druhého pravidla má toto pravidlo svá omezení, jako například nemožnost detekce při použití šifrovaného protokolu HTTPS.

Pravidlo č.5:

```
alert tcp any any -> any any (content: "www.facebook.com"; msg:
"Pristup v siti na facebook.com"; sid:10000009;)
```

Páté pravidlo detekuje pakety, jakéhokoli zdroje, jakéhokoli portu směrujícího na jakýkoli cíl jakéhokoli portu, s obsahem www.facebook.com. V tomto případě upozorní snort zprávou "Pristup v siti na facebook.com".

Pravidlo č.6:

```
alert ip any any -> 192.168.0.0/24 any (dsize: > 6000; \ msg:
"Velky IP paket nalezen";)
```

V šestém pravidle je důležité klíčové slovo dsize, které sepoužívá k nalezení délky datové části paketu. Mnoho útoků využívá zranitelnosti přetečení vyrovnávací paměti odesláním velkých paketů. Pomocí tohoto klíčového slova je možné zjistit, zda paket obsahuje data o délce větší než, menší než nebo rovno zadané velikosti. V tomto případě jde o velikost paketu větší než 6000 bajtů. Na základě tohoto pravidla IDS vypíše a zaznamená zprávu: "Velký IP paket nalezen". Dále je pravidlo omezeno na protokol IP. Zdrojový port a adresa mohou být jakékoli. Cílovou adresou může být jakákoli adresa ze subnetu 192.168.0.0./24, který lze mimojiné také prezentovat jako $HOME_NET. Takovýto zápis je také možný. Cílový port může být jakýkoli.

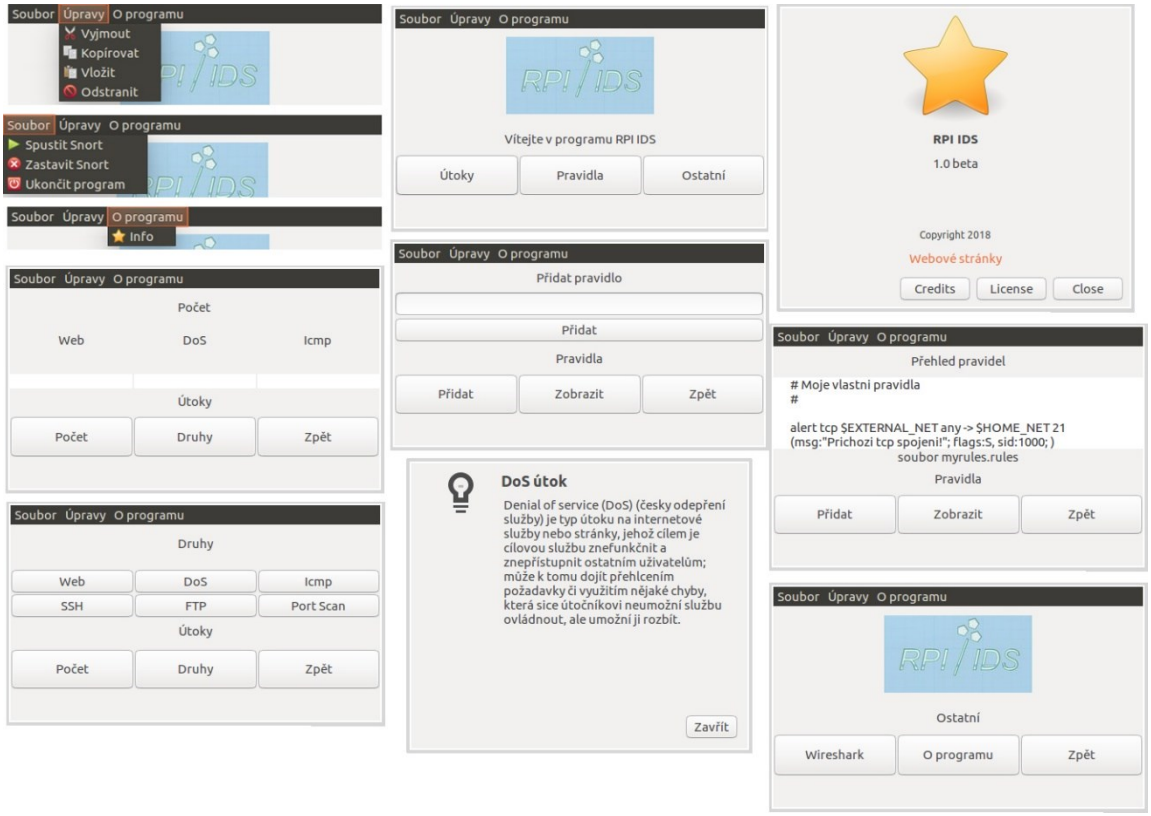
Pravidlo č.7:

```
alert tcp any any -> 192.168.0.0/24 any (flags: SF; \ msg:
"SYNC-FIN packet detected";)
```

Sedmé pravidlo detekuje každý pokus o skenování pomocí paketů SYN-FIN TCP. V případě zachycení TCP paketu od jakéhokoli zdroje jakéhokoli portu do našeho cílového subnetu obsahujícího označení SYN (argument S) a FIN (argument F) v TCP hlavičce paketu, vypíše a zaznamená zprávu o detekci s textem: "SYNC-FIN packet detected"

### 4.3.3 Tvorba návrhu vizuálního prostředí.

Po samotné realizaci IDS (kapitola 4.3) a jeho otestování (kapitola 5), došlo k návrhu vylepšení samotného IDS o koncept vizuálního prostředí. Toto prostředí by mělo mít za úkol zjednodušit přehled správci IDS a usnadnit manuální vkládání pravidel. Vizuální prostředí bylo navrženo v programu Glade s definovanými signály pro následující dodělání funkčnosti v programovacím jazyce C nebo Python. Tato programovací část dále není součástí této práce.




Obrázek 4.3: *Ukázka vytvořeného návrhu vizuálního prostředí*

# 5 Testování IDS

## 5.1 Měření výkonu RPi s implementovaným IDS

Výkon byl testován několika nástroji. Výkon při práci IDS, respektive při detekci útoků, byl měřen nástroji v odstavci níže (3.4.2). Pro ověření výkonu samotného zařízení Raspberry Pi 3 model B byl zvolen program **sysbench**. Tohoto programu bylo využito i pro Raspberry Pi 2 model B, aby mohlo dojít k vzájemnému mezigeneračnímu srovnání těchto zařízení. Menší čas = lepší výsledek.

**Výsledek sysbench testu RaspberryPi 2 model B**

```
Test execution summary:
total time:                  252.1304s
```

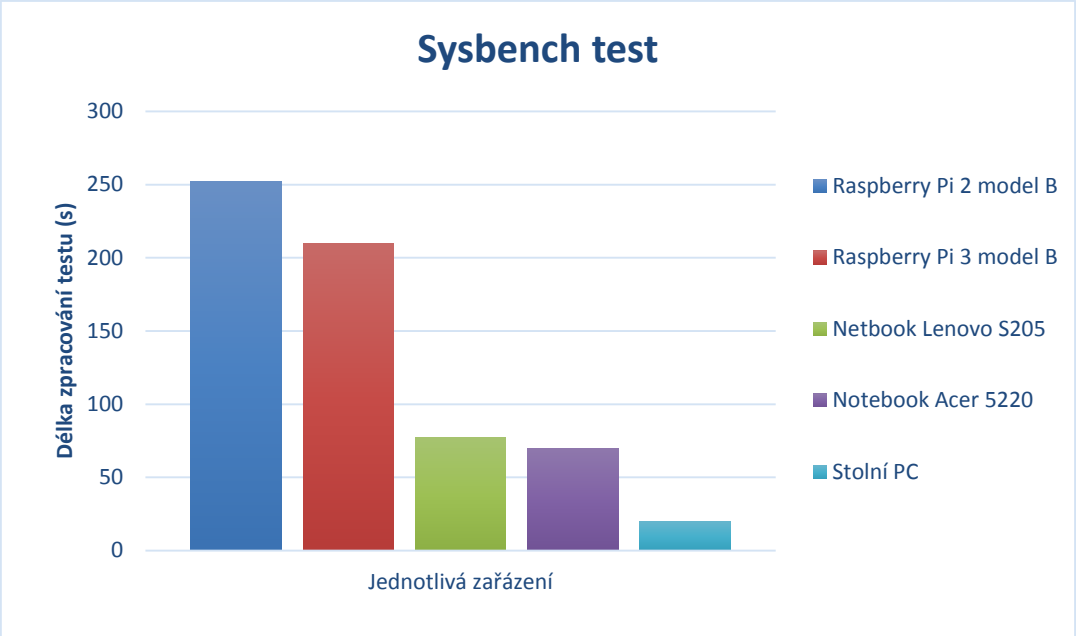
**Výsledek sysbench testu RaspberryPi 3 model B**

```
Test execution summary:
total time:                  209.8006s
```

Na základě výsledků můžeme potvrdit, že novější Raspberry Pi třetí generace dosahuje lepších výsledků než starší méně výkonnější Raspberry Pi druhé generace. Pro představu byl obdobný test proveden na netbooku, starém notebooku i výkonném PC. Výsledky k porovnání jsou uvedeny v grafu.

Tabulka 5.1: *T*abulka konfigurací jednotlivých zařízení

| *Použitá zařízení* | Raspberry Pi 2 model B | Raspberry Pi 3 model B | Netbook Lenovo S205 | Notebook Acer 5220 | Stolní PC |
|---|---|---|---|---|---|
| Procesor | ARM Cortex-A7 | ARM Cortex-A53 | AMD E-350 | Intel Celeron-M 530 | Intel Core i7-4790 |
| Paměť RAM | 1 GB | 1 GB | 2 GB | 2 GB | 16 GB |
| Grafická karta | Broadcom Videocore IV | Broadcom Videocore IV | AMD Radeon HD 6310M | Intel GMA X3100 | Nvidia GeForce GTX 960 |
| Uložiště | Micro SDXC 32 GB | Micro SDXC 32 GB | HDD 250 GB | HDD 120 GB | SSD 250 GB |
| Síťové rozhraní | 100 Mbit Ethernet | 100 Mbit Ethernet | 100 Mbit Ethernet | Gigabit Ethernet | Gigabit Ethernet |
| Operační systém | Raspbian | Raspbian | Ubuntu | Kali Linux | Kali Linux |




*Obrázek 5.1: Graf s porovnáním výsledků výkonů zařízení*

## 5.2 Testy útoků

Testovacích útoků byla zvolena řada. Od jednoduchých testovacích icmp zpráv až po DoS útoky. Během testů bylo měřeno výkonové vytížení RPi IDS pomocí nástrojů htop a sar. Htop byl využit pro aktuální sledování vytížení jednotlivých jader procesoru, paměti RAM a aktuálně běžících procesů. Sar měl za úkol sledovat a zaznamenávat vytížení procesoru a poté vypsat průměrnou procentuální vytíženostpo čas celého testu. Testy byly provedeny ve dvou zapojeních. Prvně se spojovacím prvkem v podobě směrovače. Jelikož použitý směrovač nepodporuje funkci zrcadlení provozu na portu, útoky byly směrovány přímo na zařízení s IDS. Otestováno jak na virtuálním stroji, tak v praxi. Druhé schéma mělo již nastaveno zrcadlení portu, tak aby mohlo IDS zachytávat provoz směrující z útočníkova PC na notebook oběti. Třetí zapojení bylo důležité z hlediska sdružení zrcadlení více portu a jeho dopadu na zatížení RPi IDS.

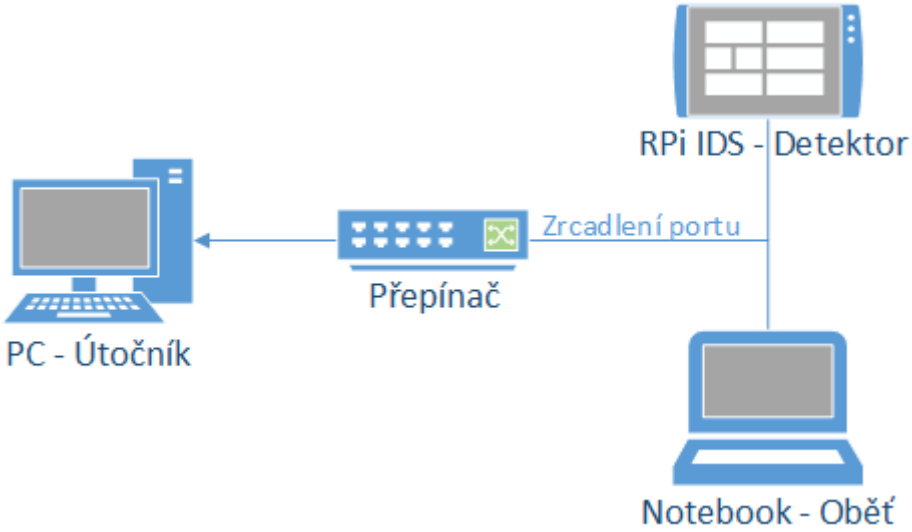
### 5.2.1 První konfigurace

V tomto prvním zapojení je obětí útoků samotný RPi IDS. PC v roli útočníka posílá jednotlivé testovací útoky na IP adresu RPi IDS. Pakety testovacích útoků takto putují z útočníka na oběť skrze směrovač.


Obrázek 5.2: *První schéma zapojení skrze router*

### 5.2.2 Druhá konfigurace

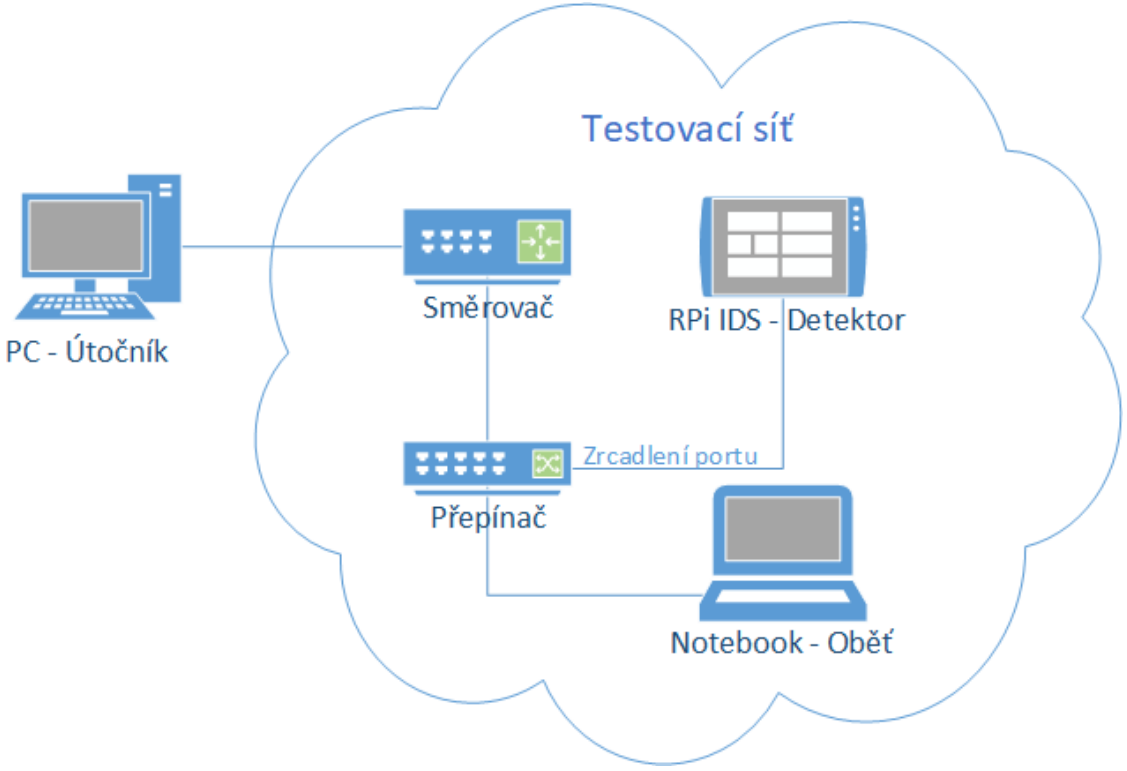
V tomto druhém zapojení bylo potřeba, aby IDS zachytával komunikaci mezi útočníkem a obětí. K tomuto účelu byl využit přepínač s funkcí zrcadlení portů (port mirroring). V případě, že PC, zde v roli útočníka, zašle testovací útok na oběť v podobě notebooku, přepínač tuto komunikaci přepošle také na port směrující k RPi IDS. IDS tak dokáže zachytit i komunikaci nepřímo směrovanou na něj.




Obrázek 5.3: *Druhé schéma zapojení skrze router*

### 5.2.3 Třetí konfigurace

Třetí zapojení je podobné druhému, nyní je však útočníkovo PC vně testovací síť a v cestě mu stojí dva síťové prvky v podobě směrovače a přepínače. Na přepínači bylo opět nastaveno zrcadlení portu tak, aby RPi IDS byl schopen zachytit komunikaci směrující na oběti. Na směrovači je nastaven překlad adres z venkovní do vnitřní testovací sítě.




*Obrázek 5.4: Třetí schéma zapojení s útočníkem mimo vnitřní síť*

## 5.3 Testování pomocí nástroje hping3

Zde bylo zvoleno celkem 11 testů od jednodušších po složitější. Dále bylo testováno, které útoky s nastavenými výchozími pravidly RPi IDS detekuje a které nedetekuje. V tabulce je dále vypsáno průměrné vytížení procesoru RPi IDS během testu.

1. Testování ICMP: V tomto příkladě se hping3 bude chovat jako běžný nástroj ping, posílat ICMP echa a přijímat ICMP odpovědi.

```
hping3 -1 (ip adresa oběti)
```

2. Traceroute pomocí protokolu ICMP: Tento příklad se podobá populárním nástrojům jako je tracert (windows) nebo traceroute (linux), který používá pakety ICMP a upravuje hodnoty TTL.

```
hping3 - traceroute -V -1 (ip adresa oběti)
```

3. Kontrola portu: Zde hping3 pošle Syn paket na zadaný port (80 v našem příkladu). Je možné také nastavit, ze kterého místního portu bude spuštěno skenování (5050).

```
hping3 -V -S -p 80 -s 5050 (ip adresa oběti)
```

4. Traceroute na určený port: Tato funkce hping3 umožňuje použíttraceroute do určeného portu a sledovat, kde je paket zablokován. To lze provést pouze přidáním traceroute k poslednímu příkazu.

```
hping3 -traceroute -V -S -p 80-s 5050(ip adresa oběti)
```

5. Další typy protokolu ICMP: Tímto příkazem bude odeslán požadavek masky adresy ICMP (ICMP addressmask - typ 17).

```
hping3-c 1 -V -1-17 (ip adresa oběti)
```

6. Jiné typy port skenování: Dále bylo provedeno skenování FIN. V připojení pomocí TCP je příznak FIN flag použit pro spuštění rutiny zavírání spojení. Pokud nedojde odpověď, znamená to, že port je otevřený. Normálně firewally posílají pakety RST + ACK zpět, aby signalizovaly, že port je zavřený.

```
hping3 -c 1 -V -p 80 -s 5050 -F (ip adresa oběti)
```

7. AckScan: Tato kontrola může být použita k zjištění, zda je hostitel dostupný (například když je Ping zablokován). To by mělo poslat odpověď RST zpět, pokud je port otevřený.

```
hping3 -c 1 -V -p 80 -s 5050 -A (ip adresa oběti)
```

8. XmasScan: Toto skenování nastaví pořadové číslo na nulu a nastaví příznaky URG + PSH + FIN v paketu. Pokud je port TCP cílového zařízení uzavřen, cílové zařízení odešle v paketu odpověď TCP RST. Pokud je otevřený port TCP cílového zařízení, cíl odstraní TCP Xmassken, aniž by odeslal odpověď.

```
hping3 -c 1 -V -p 80 -s 5050 -M 0 -UPF (ip adresa oběti)
```

9. Nulové skenování: Toto skenování nastaví pořadové číslo na nulu. Pokud je port TCP cílového zařízení uzavřen, cílový přístroj odešle odpověď TCP RST paketu. Pokud je otevřený port TCP cílového zařízení, cíl odstraní skenování protokolu TCP NULL bez odeslání odpovědi.

```
hping3 -c 1 -V -p 80 -s 5050 -Y (ip adresa oběti)
```

10. SmurfAttack: Jedná se o útok odmítnutí služby, který zaplavuje cílový systém pomocí spoofed zpráv ping vysílání.

```
hping3 -1 --flood -a (ip adresa oběti) (broadcastová adresa)
```

11. DOS Land Attack:

```
hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --flood--
rand-source (ip adresa oběti)
```

Všechny útoky byly provedeny úspěšně. Zda byl útok zachycen IDS a jaké bylo vytížení procesoru po dobu testu, lze vidět v přehledných tabulkách níže.

Tabulka 5.2: *Tabulka s výsledky útoku dle prvního schématu zapojení.*

| | Úspěšný průběh | Zachycen IDS | Vytížení CPU |
|---|---|---|---|
| | ano/ne | ano/ne | % |
| **Test 1** | ano | ano | 4.53 |
| **Test 2** | ano | ano | 4.66 |
| **Test 3** | ano | ne | 4 |
| **Test 4** | ano | ne | 2.93 |
| **Test 5** | ano | ano | 4.77 |
| **Test 6** | ano | ano | 4.25 |
| **Test 7** | ano | ne | 3.17 |
| **Test 8** | ano | ano | 4.6 |
| **Test 9** | ano | ne | 3.08 |
| **Test 10** | ano | ano | 43.56 |
| **Test 11** | ano | ano | 55.63 |

Tabulka 5.3: *Tabulka s výsledky útoku dle druhého schématu zapojení.*

| | Úspěšný průběh | Zachycen IDS | Vytížení CPU |
|---|---|---|---|
| | ano/ne | ano/ne | % |
| **Test 1** | ano | ano | 3.05 |
| **Test 2** | ano | ano | 2.72 |
| **Test 3** | ano | ne | 2.42 |
| **Test 4** | ano | ne | 2.53 |
| **Test 5** | ano | ano | 2.96 |
| **Test 6** | ano | ano | 2.88 |
| **Test 7** | ano | ne | 2.04 |
| **Test 8** | ano | ano | 2.73 |
| **Test 9** | ano | ne | 2.05 |
| **Test 10** | ano | ano | 46.6 |
| **Test 11** | ano | ano | 53.13 |

Tabulka 5.4: *Tabulka s výsledky útoku dle třetího schématu zapojení.*

| | Úspěšný průběh | Zachycen IDS | Vytížení CPU |
|---|---|---|---|
| | ano/ne | ano/ne | % |
| **Test 1** | ano | ano | 3.56 |
| **Test 2** | ano | ano | 3.34 |
| **Test 3** | ano | ne | 2.79 |
| **Test 4** | ano | ne | 3.17 |
| **Test 5** | ano | ano | 3.38 |
| **Test 6** | ano | ano | 3.37 |
| **Test 7** | ano | ne | 1.93 |
| **Test 8** | ano | ano | 3.49 |
| **Test 9** | ano | ne | 3.26 |
| **Test 10** | ano | ano | 71.97 |
| **Test 11** | ano | ne | 80.19 |

Tabulka 5.5: *Tabulka s výsledky útoku dle druhého schématu zapojení režimu sniffer*

| | Úspěšný průběh | Zachycen IDS | Vytížení CPU |
|---|---|---|---|
| | ano/ne | ano/ne | % |
| **Test 1** | ano | ano | 3.23 |
| **Test 2** | ano | ano | 3.07 |
| **Test 3** | ano | ne | 3.82 |
| **Test 4** | ano | ne | 2.78 |
| **Test 5** | ano | ano | 3.26 |
| **Test 6** | ano | ano | 3.18 |
| **Test 7** | ano | ne | 2.99 |
| **Test 8** | ano | ano | 2.78 |
| **Test 9** | ano | ne | 3.02 |
| **Test 10** | ano | ano | 75.6 |
| **Test 11** | ano | ano | 80.64 |

## 5.4 Další možné IDS

Jako další možná IDS byly zvoleny Suricata a Security Onion.

### 5.4.1 Suricata

Instalace nástroje suricata:

```
apt-get install suricata
```

K instalaci pravidel byl využit nástroj oinkmaster. V editačním programu nano byl přidán v souboru oinkmaster.conf řádek s url adresou pro instanci stažení balíčku pravidel "emerging rules".

```
nano /etc/oinkmaster.conf
```

Dále byl spuštěn nástroj oinkmaster, který automaticky stáhnul pravidla a implementoval je do příslušné složky v /etc/suricata/rules.

```
oinkmaster -C /etc/oinkmaster.conf -o /etc/suricata/rules
```

*Otestování funkčnosti IDS nástroje suricata*

Pomocí DNS překladu.

```
root@raspberrypi:/home/pi# host nsrc.org
nsrc.org has address 128.223.157.25
nsrc.org has IPv6 address 2607:8400:2880:4::80df:9d1c
nsrc.org mail is handled by 10 smtp.nsrc.org.
```

Výpisy do log souborů lze najít v /var/log/suricata. V tomto případě se výpis nacházel v souboru dns.log pomocí příkazu:

```
tail -f dns.log
```

V případě úspěšného zachycení se zobrazí následující výpis:

```
04/29/2018-13:25:10.423081 [**] Query TX 643f [**] nsrc.org [**]
AAAA [**] 192.168.0.103:49714 -> 192.168.0.1:53
04/29/2018-13:25:10.423081 [**] Response TX 643f [**] Recursion
Desired [**] 192.168.0.1:53 -> 192.168.0.103:49714
04/29/2018-13:25:10.423081 [**] Response TX 643f [**] nsrc.org
[**] AAAA [**] TTL 300 [**]
2607:8400:2880:0004:0000:0000:80df:9d1c [**] 192.168.0.1:53 ->
192.168.0.103:49714
04/29/2018-13:25:10.424103 [**] Query TX d588 [**] nsrc.org [**]
MX [**] 192.168.0.103:57159 -> 192.168.0.1:53
```

```
04/29/2018-13:25:10.424103 [**] Response TX d588 [**] Recursion
Desired [**] 192.168.0.1:53 -> 192.168.0.103:57159


04/29/2018-13:25:10.424103 [**] Response TX d588 [**] nsrc.org
[**] MX [**] TTL 10 [**] smtp.nsrc.org [**] 192.168.0.1:53 ->
192.168.0.103:57159
```

Dalším testem bylo otestování spojení pomocí icmp zpráv. Vytvořené pravidlo vypadalo následovně:

```
alert icmp any any ->$HOME_NET any (msg:"ICMP nalezeno";
sid:1000001; )
```

Test byl proveden pomocí nástroje ping dle prvního schématu zapojení z PC na RPi IDS. Pro ověření funkčnosti byl sledován,pomocí příkazu níže, výpis souboru fast.log, kde suricata ukládá log výpisy.

```
tail -f /var/log/suricata/fast.log
```

Dle výpisu níže ze souboru fast.log lze vidět, že icmp paket byl zachycen.

04/29/2018-16:56:24.297979 [**] [1:1000001:0] ICMP nalezeno [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.0.101:8 -> 192.168.0.103:0

Nástroj IDS suricata vykazoval vytížení procesoru i při nenáročném ping testu průměrné vytížení CPU na hranici 16,34%. Na tomto základě byly vybrány a odzkoušeny i náročnější testy. Konkrétně test 10 a test 11 pomocí nástroje hping3, stejně jako v předchozí kapitole 5.3.

Tabulka 5.6: Tabulka s výsledky útoku dle třetího schématu zapojení s IDS suricata

| | Úspěšný průběh | Zachycen IDS | Vytížení CPU |
|---|---|---|---|
| | ano/ne | ano/ne | % |
| Test 1 | ano | ano | 16.34 |
| Test 10 | ano | ano | 99.74 |
| Test 11 | ano | ano | 99.75 |

Jak lze vidět implementovaný IDS suricata je náročnější z hlediska požadavků na výkon. Při posledním testu došlo k samotnému zamrznutí systému RPi IDS.

### 5.4.2 Realizace IDS pomocí Security Onion

Nejprve bylo nezbytné vytvoření virtuálního stroje v programu VMware. Virtuálnímu stroji byly přiřazeny 2 jádra procesoru, 2GB paměti RAM a dvě síťové karty. Po úspěšné

instalaci OS došlo k ujištění, zda má systém Security Onion poslední akualizaci pomocí příkazu:

```
sudosoup -y
```

Dále bylo potřeba projít nastavením a instalací potřebných programů jako Sguil, Squert či Kibana.

Security Onion nabízí řadu nástrojů zastupujících roli IDS a vhodných pro monitorování sítě. Pro otestování vhodnosti implementace tohoto OS s prvky IDS byl tento systém nejprve nainstalován na virtuální stroj. K otestování byla vybrána sada útoků pomocí nástroje hping3 stejně jako při předchozím testování. Zvolené testovací nástroje IDS byly: Sguil a Squert.

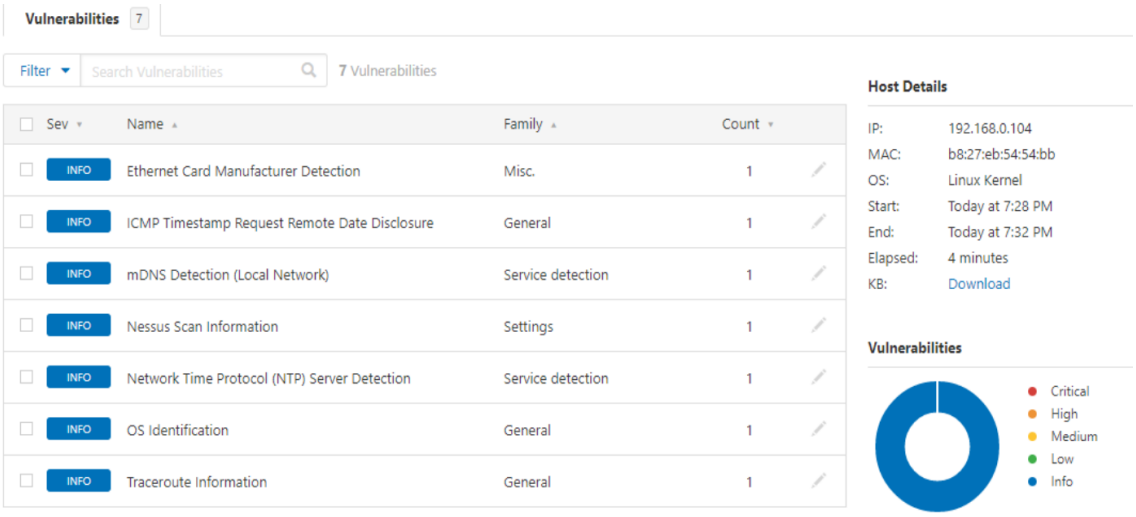
Nicméně od testováni tohoto systému se upustilo vzhledem k požadavkům na výpočetní výkon. I na virtuálním stroji s přiřazenými 2 GB paměti RAM a dvěma jádry procesoru byl systém včetně detekce útoků pomalý. Nabídl sice přehledné vizuální prostředí, a dokázal odchytit i řadu testovaných útoků ovšem jako možný OS pro RPi IDS nebyl vhodný.

## 5.5 Testování pomocí nástroje Nessus

Nástroj Nessus testuje zranitelnost systému, v tomto případě RPi IDS. Nástroj byl instalován na všechna použitá zařízení s operačním systémem Linux, ovšem na všech se při dokončování instalace, respektive při kompilaci, zasekl. Konkrétně kompilace byla tak náročná, že způsobila zamrznutí celého systému. Nakonec byla zvolena instalace Nessusu na zařízení s operačním systémem Windows 10. K instalaci je potřeba registrace u společnosti tenable, která následně poskytne uživateli aktivační klíč programu. Po úspěšné instalaci byl nástroj Nessus spuštěn a nastaven skrze webové rozhraní. V programu lze provádět mnoho možných skenů za účelem zjištění zranitelnosti. Pro otestování IDS bylo vybráno několik následujících. Jednotlivé skeny nástroje Nessus zasílají zprávy v podobě ICMP, UDP či TCP paketů a při skenovánílze sledovat, zda IDS dokáže jednotlivé skeny odhalit. Z provedených testovacích Nessus skenů dokázal IDS systém odhalit všechny.

### 5.5.1 Basic Network Scan.

Tento sken provádí 7 testů - viz obrázek níže. Dále vyhodnocuje zranitelnost dle priority.




Obrázek 5.5: *Ukázka výsledku skenu*

Pod jednotlivými testy lze po rozkliknutí vidět podrobnosti - viz obrázek 5.6.


Obrázek 5.6: *Ukázka podrobností ve výpisu zeskenu* - popis


Obrázek 5.7: *Ukázka podrobností ve výpisu ze skenu - řešení*

**Output**

```
Nessus was able to extract the following information :

  - mDNS hostname       : raspberrypi.local.

  - Advertised services :
    o Service name      : raspberrypi [b8:27:eb:54:54:bb]._workstation._tcp.local.
      Port number       : 9

  - CPU type            : ARMV7L
  - OS                  : LINUX
```

Obrázek 5.8: *Ukázka podrobností ve výpisu ze skenu - výpis*

Kromě samotného popisu (Obrázek 5.6) je možné zjistit i řešení (Obrázek 5.7) jednotlivých skenů a jejich výstupy. V tomto ukázkovém případě se jednalo o Bonjour (také známý jako mDNS) protokol, který povoluje komukoli zjistit informace vzdáleného hosta, v našem případě RPi IDS, jako jsou typ operačního systému, název hostitele, či dokonce typ procesoru. Tyto informace jsou na obrázku zobrazeny pod kolonkou Output - výstup (Obrázek 5.8).

Jak již bylo zmíněno,bylo provedeno celkem 7testů. RPi IDS reagovalo na každý z nich. Obrázkovou ilustraci s detaily jednotlivých testů lze nalézt v příloze.

## 5.6 Další možné nástroje pro testování IDS

Kromě nástroje hping3 byly použity další nástroje pro zjištění, zda IDS detekuje jednotlivé dění v síti generované vybranými nástroji.

- **ping** - nástroj na zasílání icmp zpráv a zjišťování dostupnosti cíle. *Testy tohoto nástroje dokázal RPi IDS detekovat.*
- **nmap**- nástroj pro skenování portů cíle. *Testy tohoto nástroje dokázal RPi IDS detekovat.*
- **ftp** - nástroj pro navázání ftp spojení. *Toto spojení v síti dokázal RPi IDS detekovat.*

Dalším testem na základě vytvořených testovacích pravidel bylo rozpoznání vyhledávání obsahu v síti. Například hledá-li někdo v síti slovo "terorismus", RPi IDS dokáže v zachycených paketech tuto skutečnost detekovat a nahlásit. V praxi se však detekce vyhledávaného obsahu projevila jako omezená, jelikož RPi IDS dokázalo detekovat pouze ta vyhledávání, která použila nešifrovaný protokol http. V případě, že uživatel vyhledával obsah na stránkách se zabezpečeným protokolem HTTPS, RPi IDS toto vyhledávání nebylo schopné detekovat.

### 5.6.1 Test propustnosti

Dalším vybraným testem byl zvolen test propustnosti sítě. Dle předpokladu test nedosáhl rychlosti vyšší než 100 Mbits/s, jelikož propustnost je omezena 10/100 Ethernet rozhraním samotného RPi IDS. Testy propustnosti byly využity kromě měření rychlosti také k zátěžovým testům RPi IDS.

- Měření 1 bylo provedeno skrze přepínač z notebooku Lenovo (host) na RPi IDS (server)
- Měření 2 bylo provedeno skrze přepínač a směrovač z notebooku Lenovo (host) na RPi IDS (server). Propojení mezi notebookem a směrovačem bylo realizováno pomocí rozhraní Wi-Fi ovšem s překážkou v podobě jedné cihlové zdi.
- Měření 3 bylo provedeno také skrze přepínač a směrovač z notebooku Lenovo (host) na RPi IDS (server). Propojení mezi notebookem a směrovačem bylo realizováno pomocí rozhraní Wi-Fi bez překážek.

Tabulka 5.7: Tabulka výsledných propustností jednotlivých měření

| | Úspěšný průběh | Zachycen IDS | Vytížení CPU | Propustnost |
|---|---|---|---|---|
| | ano/ne | ano/ne | % | Mbits/s |
| **měření 1** | ano | ano | 72.63 | 90.1 |
| **měření 2** | ano | ano | 22.07 | 51.2 |
| **měření 3** | ano | ano | 34.28 | 78.2 |

V případě měření propustnosti a souběžně probíhajícího útoku DoS Raspberry Pi 3 Model B vykazovalo přibližně stoprocentní vytížení.

### 5.6.2 Teplotní test

Dalším testem byl test měření teploty při zátěži na obou testovaných zařízeních. Měření byla pětiminutová a testovaná zařízení po celou dobu testu byla vytížena sysbench procesem. Výsledná měření byla zprůměrována a jejich výsledky vzájemně porovnány.

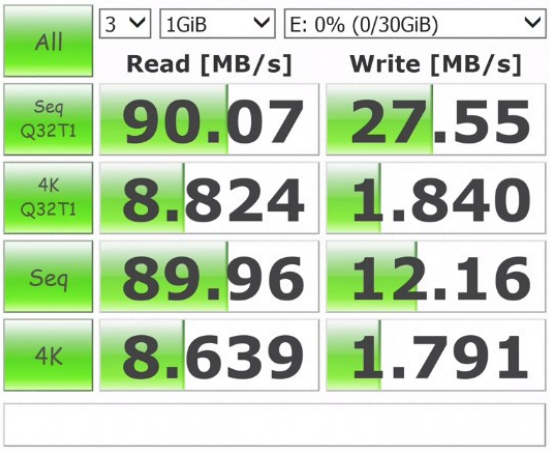
*Průměrná teplota RaspberryPi 2 model B v zátěži: 62,32°C*

*Průměrná teplota RaspberryPi 3 model B v zátěži: 75,51°C*

Z těchto výsledků lze usoudit, že u novější generace s nárůstem výkonu vyrostla i teplotní náročnost. Na základě těchto testů lze doporučit nasazení chladicího prvku pro lepší odvod tepla komponentu RaspberryPi.

### 5.6.3 Test rychlosti úložiště

Raspberry Pi 3 Model B využívá pro běh systému microSD karet. Karty mohou mít velkou kapacitu, avšak jejich určitou limitací mohou být rychlosti čtení a zápisu. Zvolenou kartou pro uložení a běh systému byla Samsung SDXC Evo Plus 32GB. Kapacita karty je dostačující a vlastnosti jako rychlosti zápisu a čtení by podle dostupných zdrojů měly být také dostačující. Přesto byl proveden test samotné karty bez systému v programu CrystalDiskMark na prázdné paměťové kartě bez nainstalovaného OS.




*Obrázek 5.9: Výsledky testu rychlosti MicroSD karty Samsung Evo Plus 32GB*

Další test rychlosti paměťové karty byl proveden, na již běžícím systému Raspbian. Jak lze vidět na obrázku 5.10. rychlost čtení byla 16.1.MB/s a pro zápis 7MB/s. Rychlosti jsou tedy podstatně nižší než byly v případě měření prázdné paměťové karty.

```
root@raspberrypi:/home/pi# dd if=~/test.tmp of=/dev/null bs=500K count=1024
1024+0 records in
1024+0 records out
524288000 bytes (524 MB) copied, 32.498 s, 16.1 MB/s
root@raspberrypi:/home/pi# dd if=/dev/zero of=~/test.tmp bs=500K count=1024
1024+0 records in
1024+0 records out
524288000 bytes (524 MB) copied, 74.9434 s, 7.0 MB/s
```

Obrázek 5.10: *Výsledky testů rychlosti zápisu a čtení na samotném RPi IDS*

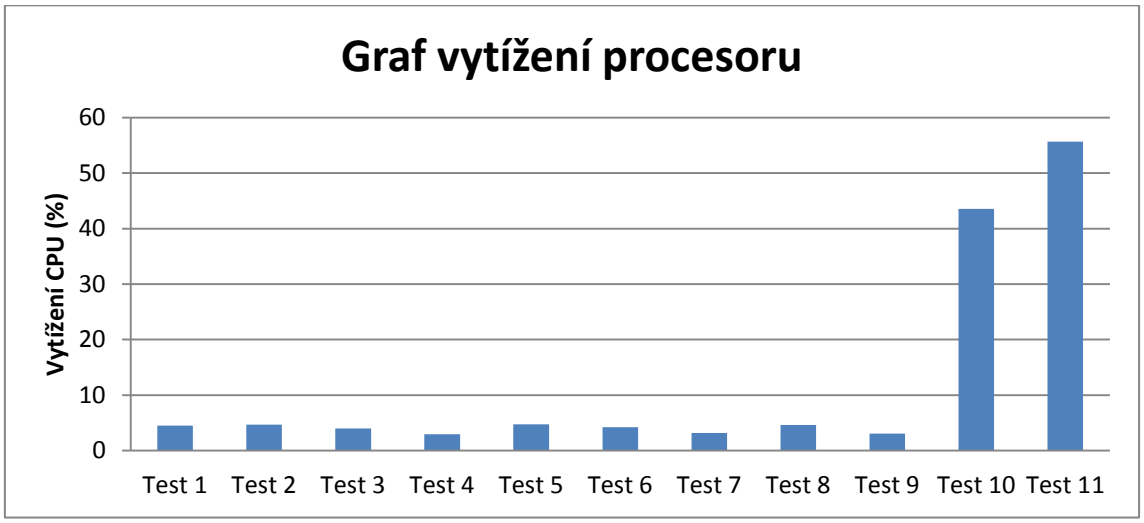
## 5.7 Vyhodnocení testů

Na základě provedených testů lze posoudit, že výkon testovaného Raspberry Pi 3 je vyšší pouze o 20,176%. Výsledek je ovlivněn softwarovou nadstavbou. V tomto případě tedy nelze souhlasit s tvrzením výrobce, že došlo až k 30% nárůstu výkonu oproti předchozí generaci Raspberry Pi, alespoň co se výkonu procesoru týče.

Pomocí testů se zjistilo, že Raspberry Pi 3 model B trpí při zátěži větším zahříváním komponentů na základní desce než předešlá generace. Počítá-li uživatel s náročnějším využíváním tohoto zařízení, doporučuje se jej opatřit aktivním, nebo alespoň pasivním chlazením.

Testy detekce IDS odhalily, že s nárůstem provozu v síti stoupají nároky na výkon IDS systému. Dochází tak k většímu výkonnostnímu vytížení samotného zařízení s implementovaným IDS. RPi IDS se projevilo jako dostatečné pro malé sítě o velikosti maximálně dvou hostů. Většinu testů RPi IDS zvládlo a bylo by schopné pracovat i s více hosty v síti. Ovšem u některých testů se projevily výkonnostní nedostatky. Nejvíce se to projevilo při

navýšení počtu útoků. Při testování dvou souběžných DoS útoků docházelo ke zpoždění detekcí útoků. V případě tří souběžně běžících DoS útoků došlo až k zamrznutí samotného systému RPi IDS. V grafu průměrného vytížení ze všech 3 možných zapojení také potvrzuje, že poslední dva testy byly podstatně náročnější na výkon IDS systému. Většinu ostatních testů však dokázal RPi IDS detekovat, aniž by došlo k zásadnějšímu výkonovému vytížení. Sada vytvořených pravidel community-rules od tvůrců snortu stažená z oficiálních stránek se ověřila dobře, a většinu útoků dokázala odhalit.




Obrázek 5.11: Grafprůměrného vytížení procesoru jednotlivých testů při všech třech zapojeních

Testovaný systém SecurityOnion se projevil jako vhodný pro implementaci IDS. Kromě spolehlivé detekce útoků nabízí i řadu nástrojů s přehledným vizuálním prostředím. Přestože měl tento systém přiřazena dvě jádra procesoru a 2GB paměti RAM ve virtuálním stroji, jeho běh nebyl zcela plynulý. Doporučuje se výkonnější procesor a jako minimum alespoň 3GB paměti RAM. Není tedy vhodný pro implementaci na RaspberryPi.

# Závěr

Cílem této práce bylo sestrojení a ověření IDS analyzátoru na platformě Raspberry Pi. První část práce se zabývá nastíněním problematiky a příslušné teorie k ní. Dále rešerší jednočipových počítačů možných pro tuto realizaci. Na základě teoretických poznatků bylo potřeba přenosný IDS analyzátor sestrojil. Po výběru vhodných komponent byl vytvořen návrh obalu pro 3D tisk. Návrh obalu byl proveden pomocí programu 123D Design. Dále byly provedeny nezbytné kroky pro samotný 3D tisk. Jako vhodná pro tisk byla zvolena školní 3D tiskárna Prusa3D MK2. Jednotlivé komponenty byly poté implementovány do vytištěného plastového obalu. Dále byly na sestavené handheld zařízení nainstalovány operační systém a nástroje pro další testování.

Dalším cílem práce byla implementace systému detekce průniku IDS. Za vhodnýnástroj pro realizaci IDS byl zvolen nástroj Snort, který dokáže analyzovat a detekovat nežádoucí obsah v síti na základě definovaných pravidel. Pravidla byla použita, jakpředddefinována od tvůrců tohoto nástroje, tak i vlastní vytvořená v souboru myrules.rules (Příloha B.). Hlavními body implementace IDS bylo správné nastavení souboru snort.conf a implementace pravidel nacházejících se ve složce rules. Pro nastínění zjednodušení práce s tímto nástrojem byl dále vytvořen návrh vizuálního prostředí IDS.

Následovalo otestování implementovaného IDS systému na platformě Raspberry Pi formou generovaných síťových útoků a penetračních testů s ohledem na výkonnost testovaného zařízení. Měřila se tedy jak úspešnost detekce jednotlivých útoků, tak výkonové vytížení RPi IDS při detekci paketů. Ukázalo se, že Raspberry Pi 3 Model B má dostatečný výkon pro jednodušší testy, ovšem v případě náročnějších testů, jako například DoS útoků, se ukázalo, že výkon zařízení je nedostačující. Také v případě několika souběžně běžících útoků, nebo například souběžně běžícího testu propustnosti a probíhajícího testovacího útoku, RPi IDS vykazovalo téměř stoprocentní vytížení CPU a paměti RAM. S narůstajícím rozsahem sítě stoupají i požadavky na výkon IDS systémů, proto je nutno podotknout, že testované zařízení v roli IDS systému může být vhodné analýzu jednoho hosta (oběti). Pro větší sítě, počet hostů, není použití RPi IDS vhodné z důvodu nedostatečného výkonu zařízení. Jako dalším otestovaným nástrojem pro funkci IDS byla Suricata. Suricata se projevila obdobně spolehlivá z pohledu detekce, jako předchozí Snort, ovšem její náročnost na výkon se byla při testování podstatně vyšší, tím pádem byla vyhodnocena jako méně vhodná pro implementaci IDS na platformě Raspberry Pi. Testy výkonu zařízení, teploty či rychlosti microSD karty odhalily další nedostatky této platformy pro užití pro IDS. Toto zařízení vhodné pro experimentální testování či jako výukový nástroj do škol.

Zaujala mne rozmanitost testu/útoků, které testovaný IDS systém dokáže detekovat, a možnosti úprav jednotlivých pravidel správcem IDS na míru. Toto testování IDS mne zaujalo natolik, že v něm budu pokračovat nadále i po této práci, ovšem na výkonnějším hardwaru a rozsáhlejší síti.

# Použitá literatura

[1] BEALE, Jay, Andrew R. BAKER a Joel. ESLER. Snort: IDS and IPS toolkit. Burlington, MA: Syngress, c2007. ISBN 978-1597490993.

[2] REHMAN, Rafeeq Ur. Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID. Upper Saddle River, N.J.: Prentice Hall PTR, c2003. ISBN 0131407333.

[3] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. 2., aktualizované vydání. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 9788025148198.

[4] Bezpečnost sítí: velká kniha. Brno: CP Books, 2005. ISBN 9788025106976.

[5] HONTAÑÓN, Ramón J. Linux: praktická bezpečnost. Praha: Grada, 2003. ISBN 9788024706528.

[6] LUDVÍK, Miroslav a Bohumír ŠTĚDROŇ. Teorie bezpečnosti počítačových sítí. Kralice na Hané: Computer Media, 2008. ISBN 80-86686-35-3.

[7] PROSISE, Chris a Kevin MANDIA. Počítačový útok: detekce, obrana a okamžitá náprava. Praha: Computer Press, 2002. Komunikace a sítě. ISBN 8072266829.

[8] Raspberry Pi - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 27.04.2018]. Dostupné z: https://www.root.cz/n/raspberry-pi/

[9] IDS Snort [online]. Radomír Orkáč, 2006 [cit. 2018-04-27]. Dostupné z: http://www.cs.vsb.cz/grygarek/SPS/projekty0506/Snort.pdf

[10] Raspberry Pi 3 Model B 64-bit 1GB RAM - RPishop.cz. RPishop.cz [online]. Dostupné z: http://rpishop.cz/kategorie/283-raspberry-pi-3-model-b-64-bit.html

[11] Co je to Arduino? | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice [online]. Dostupné z: https://arduino.cz/co-je-to-arduino/

[12] Raspberry Pi. RPishop.cz [online]. Dostupné z: http://rpishop.cz/raspberry-pi-pocitace/170-raspberry-pi-2-1024-mb-ram.html

[13] Co je to Arduino? | Arduino.cz. Arduino.cz - Webový magazín o Arduinu a elektronice [online]. Dostupné z: https://arduino.cz/co-je-to-arduino/

[14] Seriál Banana Pi R1 - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 27.04.2018]. Dostupné z: https://www.root.cz/serialy/banana-pi-r1/

[15] Libor Marek výuka na CMGaSOŠPg Brno [online]. Dostupné z: http://www.marlib.cmsps.cz/os/os.html

[16] Ubuntu - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 27.04.2018]. Dostupné z: https://www.root.cz/n/ubuntu/

[17] Bezpečnost IS/IT. Mendelu.cz [online]. Brno [cit. 2018-04-27]. Dostupné z: https://akela.mendelu.cz/~lidak/bis/7tech.htm

[18] Snort rules - pravidla snortu [cit. 28.04.2018]. Dostupné z: https://snort.org/downloads/#rule-downloads

[19] Nessus Professional Data Sheet [online]. Copyright © [cit. 28.04.2018]. Dostupné z: http://info.tenable.com/rs/934-XQB-568/images/NessusPro__DS__EN_v8.pdf

[20] Suricata | Open Source IDS / IPS / NSM engine. Suricata | Open Source IDS / IPS / NSM engine [online]. Dostupné z: https://suricata-ids.org

[21] Suricata - Open Source Next Generation IDS/IPS Engine. CyberPunk: The Best Open Source CyberSecurity Tools [online]. Dostupné z: https://n0where.net/open-source-ips-suricata

[22] Security Onion. Security Onion [online]. Copyright © Security Onion Solutions, LLC [cit. 30.04.2018]. Dostupné z: https://securityonion.net

# Seznam příloh

Součástí BP/DP je CD/DVD.

Adresářová struktura přiloženého CD/DVD:

Vizuální prostředí/GUIrpi structure.jpg

Vizuální prostředí/Navrh GUI.glade

3D tisk/rpi ids komplet.123dx

Pravidla/myrules.txt